

Adaptive Resource Management Technology for Satellite Constellations

Lonnie R. Welch and Brett Tjaden

School of EECS
Ohio University
Athens, OH
welch@ohio.edu

Barbara B. Pfarr

Goddard Space Flight Center
Greenbelt, MD
bpfarr@pop500.gsfc.nasa.gov

Abstract—This manuscript describes the Sensor Web Adaptive Resource Manager (SWARM) project. The primary focus of the project is on the design and prototyping of middleware for managing computing and network resources in a way that enables the information systems of satellite constellations to provide real-time performance within dynamic environments. The middleware has been prototyped, and it has been evaluated by employing it to manage a pool of distributed resources for the ITOS (Integrated Test and Operations System) satellite command and control software system. The design of the middleware is discussed and a summary of the evaluation effort is provided.

I. INTRODUCTION

The computer resource requirements for satellite constellations (SCs) will be much greater than for the satellite systems of today [1, 2]. To best meet those requirements, SCs should have available a pool of computing resources that are distributed across satellites and ground stations. Currently, we do not know how to harness the distributed computational and communication resources that likely will be available to SCs, which exposes the SC initiative to the following risks: (1) poor performance, resulting in operators, engineers and scientists seeing stale data and learning about critical events after they occur, (2) missed opportunities to process important terrestrial events, and (3) inefficient use of resources.

To mitigate these risks it is important to answer several questions. How will SCs be operated? What will the onboard information technology systems of SCs be like? How can a pool of computing and network resources, that exist both on the SCs and on the ground, be used to effectively support the onboard information technology systems?

Thus, the Sensor Web Adaptive Resource Manager (SWARM) project [6, 9] is investigating how SCs can exploit the opportunities of distributed computing. We are developing computing and network resource management middleware (software which resides between application programs and operating systems and provides services to application programs) to enable “distributed computing in-the-sky” (see Section II); studying SC concepts and the space systems of today to determine the properties of information systems of satellite constellations; and producing a realistic ground-based SC

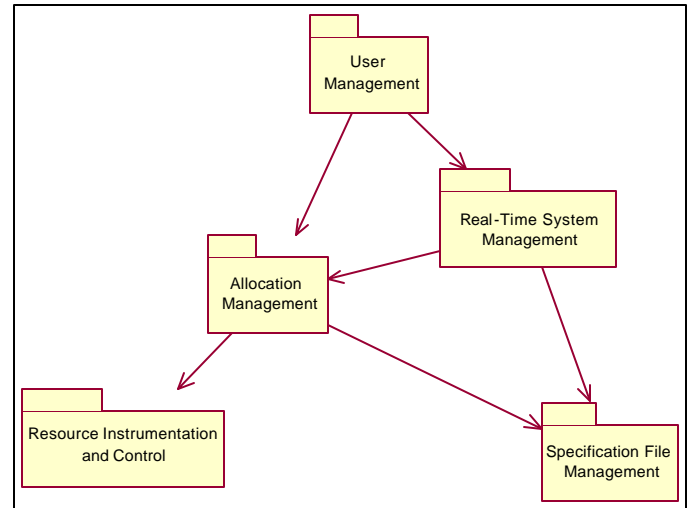


Figure 1. The major subsystems of the middleware and their dependencies.

command and control testbed to demonstrate and evaluate the technology (see Section III).

II. COMPUTING AND NETWORK RESOURCE MANAGEMENT TECHNOLOGY

The primary focus of our effort is on the design and prototyping of middleware for managing the computing and network resources in a way that enables the information systems of satellite constellations to provide real-time performance within dynamic environments [4]. The middleware has been developed and designed using Rational’s unified process and described using the unified modeling language (UML) [10].

The architecture (see Figure 1) of our resource management (RM) middleware consists of five major subsystems: User Management, Allocation Management, Real-Time System Management, Resource Instrumentation and Control, and Specification File Management.

Specification File Management parses hardware configuration and software specification files. The specification files describe the characteristics of the computing and network resources and the features and real-time requirements of the information system software [7, 8].

The Resource Instrumentation and Control subsystem has two main purposes. First, its Resource Monitor component is used to gather information about the utilization and availability of the computing and network resources [3, 7]. Second, its Application Control component is used to start and stop the application programs that constitute the information system software.

The User Management subsystem allows an operator of the RM middleware to command the Allocation Manager to start or stop a real-time system. It also allows the RM operator to view a real-time system's performance.

The Real-Time System Management subsystem monitors the performance of real-time systems and provides updates to the User Management subsystem (communication among middleware components is done using CORBA [5]). When real-time performance problems are detected, Real-Time System Management performs diagnosis of the causes, identifies possible resource reallocation actions that could be taken to restore required real-time performance, and reports its findings to the Allocation Management subsystem.

Allocation Management uses Resource Instrumentation and Control to (1) gather information about the resources, and (2) start and stop application programs. The resource information is used to maintain a feasible allocation (one in which all real-time requirements are met) and that provides optimal utility to all information systems under its control.

The most critical use case of the middleware system, *maintain a feasible allocation*, is illustrated in Figure 2. Real-time systems report their performance data to Real-Time System Management, which monitors real-time performance and requests that Allocation Management reallocate resources if a real-time performance problem is detected. Allocation Management creates a reallocation plan and uses Resource Instrumentation and Control to execute the plan.

One of the technological innovations of the project is the development of middleware to monitor, analyze, and control network resources. We have developed a network monitoring program (see Figure 3) that reports the amount of bandwidth being used between pairs of hosts. In order to obtain this information, the monitoring program must know the topology of the networks and interconnections that comprise the system. We have developed hardware configuration specification language constructs for describing network-related information such as hosts, network devices, network interfaces, and network connections [7]. Our network monitoring program obtains the topology and connectivity of the real-time system from a specification file at startup time. The network monitor periodically uses the Simple Network Management Protocol (SNMP) to gather performance information from hosts and network devices. Combining the SNMP query results and

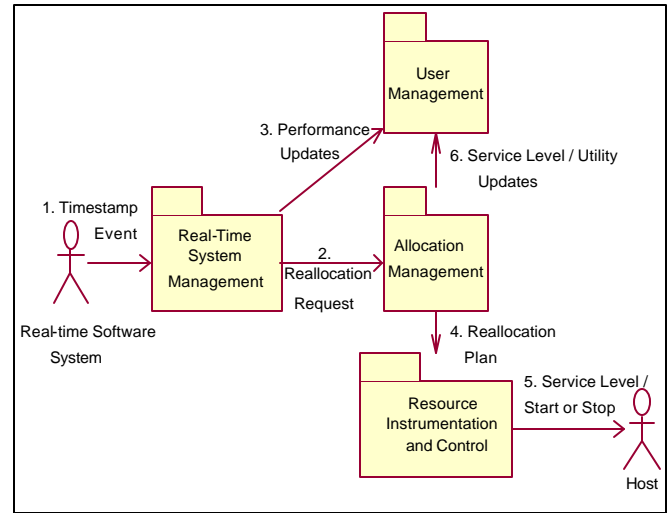


Figure 2. The subsystem collaboration diagram for the *maintain feasible allocation* use case.

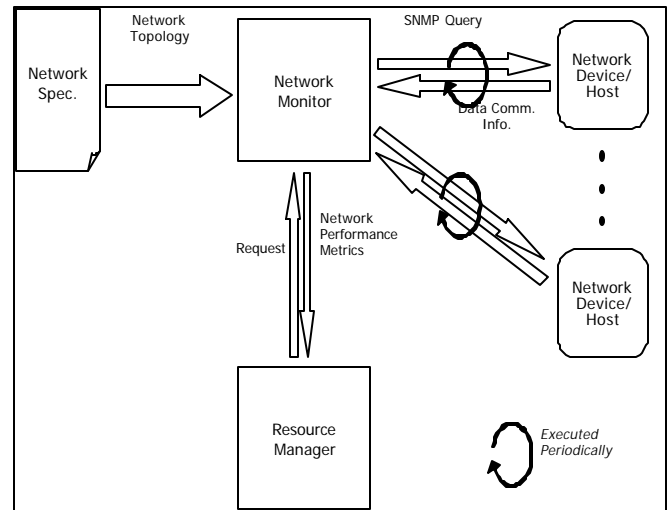


Figure 3. The network monitoring middleware.

network topology information, bandwidth statistics are calculated and reported on demand.

The RM middleware will enable a pool of computing and network resources of SCs to be *shared* and to be reallocated dynamically in response to important events.

Figure 3 illustrates what is possible with adaptive RM technology. Multiple satellites can provide data, which is processed by various software systems (such as science processing, command management, mission planning, and flight dynamics). Even though the mix of software systems may change and the amount of processing and inter-process communication for a particular set of software systems may vary, the RM middleware will maintain an allocation of resources that allows the real-time requirements to be met.

Our adaptive RM middleware provides innovations not found in related approaches. Traditional load balancers (such as Mosix) and resource scavengers (such as Condor) do not

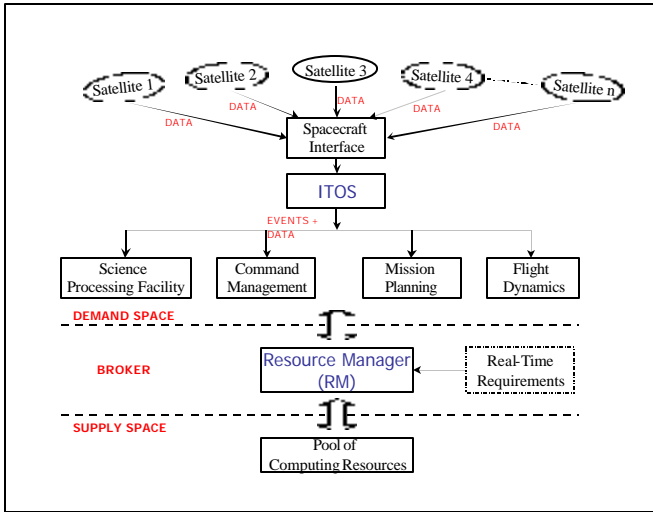


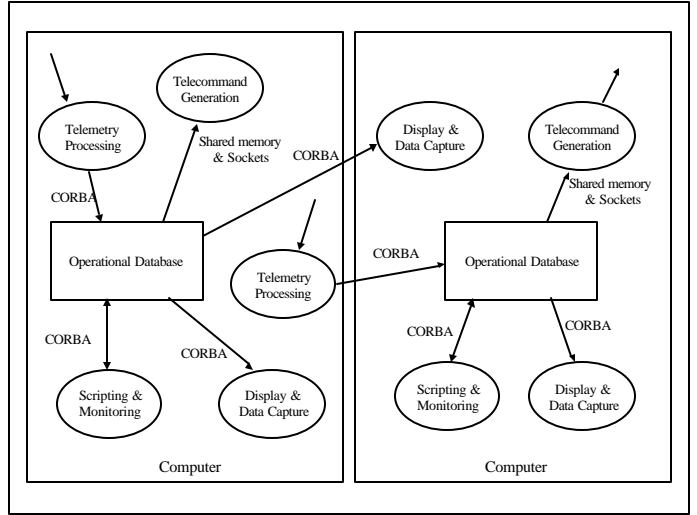
Figure 4. Resource management of a pool of computing resources in support of the information systems for satellite constellations.

consider real-time constraints. Traditional real-time resource allocation and scheduling approaches (such as rate monotonic analysis) cannot accommodate dynamic real-time systems and manage only the computing resources (not the network resources). Our approach overcomes all of these shortcomings.

III. TECHNOLOGY DEMONSTRATION AND EVALUATION

The RM technology has been prototyped and has been evaluated by employing it to manage a pool of distributed resources for the ITOS (Integrated Test and Operations System) satellite command and control software system.

ITOS (see Figure 4) is a telemetry and command system for spacecraft monitoring and control. It provides telemetry



processing, capture, and display; telecommand generation; and scripting and monitoring capabilities.

Figure 5. Distribution of the major components of ITOS between two computers.

One of the accomplishments of this project has been the resource management of the ITOS system. ITOS was originally designed to be statically allocated to a single computer. We have successfully decomposed the ITOS software into components that run on different host computers and have inserted code that reports the real-time performance of ITOS to the RM middleware. The RM middleware performs the following services for ITOS: automatic starting and stopping, instrumentation of real-time performance, distribution of software components among multiple hosts, survivability, detection of real-time performance problems, and automatic reallocation of ITOS components to restore real-time performance.

IV. CONCLUSIONS AND FUTURE WORK

This paper has described an effort that is developing middleware for management of a pool of computing resources for satellite constellations, determining likely characteristics of satellite constellation information systems, and producing a distributed, ground-based satellite command and control testbed.

The middleware has been designed and prototyped. Furthermore, a distributed satellite command and control testbed has been developed and used to evaluate the middleware. The effectiveness of the middleware for managing a pool of resources to provide real-time performance and survivability for the ITOS system has been demonstrated.

This work is beneficial to NASA for several reasons. It provides middleware technology that reduces the risk of network-computing-in-space in the following ways. A new concept of fault tolerance is possible, whereby failed processes can be quickly restarted by the middleware. There is also reduced risk because processes are not statically assigned to resources (which may fail), but may run on any available processor, with backups available on other processors. Ability to handle unknown environments also increases, because the middleware can run copies of one process simultaneously on multiple processors as a way of load sharing.

Additional benefits accrue to NASA from the application of the technology to a NASA satellite command and control system.

The ability to automatically restart failed processes and the flexibility provided for controlling the processing associated with telemetry displays is potentially useful for many projects and applications. Furthermore, because of this initiative, the developers of the ITOS system have considered radical changes in their software architecture. Additionally, the use of a complex operational system has led to identification of new requirements for resource management middleware.

The production of a ground-based testbed for exploration of reconfigurable information technology for satellite constellations is an important contribution because the performance of "adaptive" spacecraft systems is still unknown and needs to be investigated. The testbed provides basic telemetry and command functions that are under control of a distributed, adaptive middleware system; other applications (e.g., planning and scheduling, data processing, and science processing) can be added to complete an on-orbit architecture prototype.

The utility of the adaptive resource management middleware prototype for the information systems of distributed satellite systems has been shown. Future plans include the advancement of the technology to further enhance usability of

the RM middleware for space environments (e.g., managing threads and aperiodic software components, and automatically generating specification files). Finally, we plan to enhance the testbed with mission applications and with simulated instruments.

V. REFERENCES

- [1] G. E. Prescott, S. A. Smith and K. Moe, "Real-time information system technology challenges for NASA's Earth Science Enterprise," in *Proceedings of the International Workshop on Real-Time Mission-Critical Systems*, Dec. 1999.
- [2] S. Jain, L.R. Welch, D.M. Chelberg, Z. Tan, D. Fleeman, D. Parrott, B. Pfarr, M.C. Liu, and C. Shuler, "A collaborative problem solving agent for on-board real-time systems," *The 10th Workshop on Parallel and Distributed Real-Time Systems*, April 2002.
- [3] Hong Chen, Brett Tjaden, Lonnie Welch, Carl Bruggeman, Lu Tong, Barbara Pfarr, "Monitoring network QoS in a dynamic real-time system," *The 10th Workshop on Parallel and Distributed Real-Time Systems*, April 2002.
- [4] Lonnie R. Welch, Paul V. Werme, Behrooz A. Shirazi, Charles D. Cavanaugh, Larry Fontenot, Eui-Nam Huh and Michael W. Masters, "Load balancing for dynamic real-time systems," *Cluster Computing*, **3(2000)**:125-138, 2000.
- [5] S. Anwar and L. Welch, "Experience with TAO for development of adaptive resource management middleware," 1st Workshop on The ACE ORB (TAO), August 2001.
- [6] Toni Marinucci, Anbuselvan Neelamegam, Brett Tjaden, Lu Tong, Lonnie Welch, Brian Goldman, Greg Greer, Deepak Kaul and Barbara B. Pfarr, "Sensor web adaptive resource manager," *NASA Earth Science Technology Conference (ESTC-2001)*, August 2001.
- [7] Lu Tong, Carl Bruggeman, Brett Tjaden, Hong Chen, Lonnie R. Welch, "Specification and modeling of network resources in dynamic, distributed real-time systems," *ISCA 14th International Conference on Parallel and Distributed Computing Systems*, August 2001.
- [8] A. Neelamegam, C. Bruggeman, and L. Welch, "Modeling multiple and heterogeneous streams for resource management," *2001 International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA 2001)*, 1434-1438, June 2001.
- [9] Ryan Detter, Lonnie R. Welch, Barbara Pfarr, Brett Tjaden, and Eui-Nam Huh, "Adaptive management of computing and network resources for spacecraft systems," *The 3rd Annual Military and Aerospace Applications of Programma-*

ble Devices and Technologies Conference (MAPLD 2000),
September 2000.

[10] I. Jacobson, G. Booch and J. Rumbaugh, *The Unified
Software Development Process*, 1999, Addison Wesley
Longman, Inc